TELKOMNIKA Telecommunication Computing Electronics and Control
Vol. 23, No. 3, June 2025, pp. 807~815
ISSN: 1693-6930, DOI: 10.12928/TELKOMNIKA.v23i3.24994

Indonesian continuous speech recognition optimization with convolution bidirectional long short-term memory architecture

Sukmawati Nur Endah, Rismiyati, Priyo Sidik Sasongko, Anwar Petrus F. Naiborhu
Department of Informatics, Faculty of Science and Mathematics, Universitas Diponegoro, Semarang, Indonesia

Article Info

Article history:

Received Mar 13, 2024
Revised Feb 25, 2025
Accepted Mar 11, 2025

Keywords:

Bidirectional long short-term memory
Continuous speech
Convolution bidirectional long short-term memory
Indonesian speech recognition
Speech recognition

ABSTRACT

Speech recognition can be defined as converting voice signals into text or lines of words by using algorithms implemented in computer programs. There are several types of speech recognition, including recognition for isolated word speech, continuous speech, spontaneous speech, and conversational speech. Research on continuous speech recognition, especially in Indonesian, has been developed using both stochastic methods such as Hidden Markov model (HMM) and deep learning methods. Currently, deep learning approaches are more widely used in speech recognition applications. This research optimizes Indonesian speech recognition by adding convolution layers to the bidirectional long short-term memory (Bi-LSTM) architecture. The goal of this research is to find the best architecture so that better Indonesian continuous speech recognition results can be obtained. The dataset used in this research was created by the intelligent systems research group in the Department of Informatics at Universitas Diponegoro. All speakers who participated in this dataset came from five ethnic groups in Indonesia, representing the dialects of their respective ethnic groups. The research results show that by adding a convolution layer to the Bi-LSTM architecture, speech recognition performance increases significantly with an average word error rate (WER) reduction of 15.56% compared to using only the Bi-LSTM architecture.

Corresponding Author:

Sukmawati Nur Endah
Department of Informatics, Faculty of Science and Mathematics, Universitas Diponegoro
Prof Soedarto, S.H. Street, Campus of Tembalang UNDIP, Semarang, Central of Java, Indonesia
Email: sukmane@lecturer.undip.ac.id

1. INTRODUCTION

Speech recognition is part of the speech processing research area which in recent years has been widely developed in various application fields such as education, health, telecommunications and entertainment. It can be defined as the process of converting speech signals into text or lines of words by using machine learning algorithms and implemented in computer programs [1]. There are several types of speech recognition, including isolated word speech recognition [2], continuous speech [3], and spontaneous speech [4].

Research on continuous speech recognition, especially for Indonesian, has been developed using both stochastic methods such as Hidden Markov model (HMM) [5], [6] and deep learning [7]-[11]. Deep learning techniques have been widely used in recent years, enabling increased accuracy in automatic speech recognition (ASR) systems like the research of [12], [13]. The use of deep learning aim to simplify the processing pipeline by training a single model to directly map speech signals to their corresponding text transcriptions [1]. Most end-to-end ASR systems use deep neural networks (DNNs) to learn acoustic and linguistic representations directly from the input speech signals [14]. Deep learning techniques in ASR are

always developing, starting from recurrent neural network (RNN) [15], [16], long-short time memory (LSTM) [17], [18], bidirectional long short-term memory (Bi-LSTM) [19], Transformer [20], [21] to Conformer [22], [23]. This development emerged to overcome various weaknesses of previous methods. For example, LSTM overcomes the weakness of RNN which can cause vanishing gradients. Conformer is a development of Transformer by adding a convolution block to the transformer encoder. For ASR tasks, this can reduce the word error rate (WER) value.

Compared to traditional acoustic models, RNN-based architectures offer several advantages in speech recognition. One key benefit is their ability to capture long-term temporal dependencies [24], [25] in speech data, allowing them to model the dynamic characteristics of speech signals effectively. However, RNNs struggle with processing long sequences due to memory limitations and the vanishing gradient problem [26].

A solution to this issue is LSTM, which significantly improves RNNs' ability to learn long-term dependencies [27]. LSTM has proven effective in speech recognition tasks [28]. However, a limitation of conventional RNNs, including unidirectional LSTM, is that they only utilize past context. In speech recognition, where entire speech sequences are transcribed simultaneously, leveraging future context is also beneficial. Bidirectional RNN (BRNN) [29] addresses this by processing data in both directions using two hidden states, which are then passed to the same output layer. Combining BRNN with LSTM results in Bi-LSTM, allowing the model to utilize the past and future contexts for better learning.

Bi-LSTM networks have been applied to phoneme classification task [30], [31], with studies showing that they outperform unidirectional LSTM and standard RNNs in frame-wise phoneme classification. This suggests that Bi-LSTM is well-suited for speech processing, where contextual information plays a crucial role [26]. However, recent research indicates that certain convolutional neural network **(**CNN) architectures can achieve state-of-the-art accuracy in tasks like audio synthesis, word-level language modeling, and machine translation [32], [33]. CNNs offer the advantage of faster training through parallel computation. It can also overcome common challenges associated with recurrent models, such as the vanishing or exploding gradient problem and difficulties retaining long-term memory [1]. Therefore, this research optimizes Indonesian speech recognition by adding a convolution layer to the Bi-LSTM architecture. The research aims to find out whether adding convolution to several Bi-LSTM layers will be able to reduce the WER so that it can improve the performance of continuous speech recognition in Indonesian.

2. RESEARCH METHOD

The research consists of several stages: data collection, dataset distribution, feature extraction, normalization, model training, speech recognition and evaluation.

2.1. Data collection

The dataset used in this study is the same as that used in [7] which collection of speech from several types of dialects from ethnic groups in Indonesia. This dataset was produced solely by the Department of Computer Science/Informatics, Universitas Diponegoro. All of the speakers who took part in the creation of this dataset came from all over Indonesia by representing the dialect of each speaker's tribe. There are five types of dialects used in this dataset, namely Balinese, Batak, Javanese, Minang, and Sundanese dialects. There are 23 speakers involved in the data acquisition process. The recording process used Audacity software with a sample rate of 44,100 Hz, 32-Bit Float type, mono channel and each audio was stored in wav format (waveform audio file format).

There are 70 sentences spoken in this dataset. The sentences that the speaker uttered consisted of 20 sentences from our beginning research about speech recognition and 50 sentences from previous research about disaster mitigation. However, only 19 speakers uttered 70 sentences while the others only uttered 20 sentences from the beginning research. These sentences are structured randomly. Sentences related to disaster mitigation are structured as closely as possible to the situation in the field. The number of sentences spoken by 23 speakers is 1410.

2.2. Data splitting

From the data that has been collected, the data is split into training, validation and testing data. The distribution of data for training data, validation data and test data are 60%, 20%, and 20 %, respectively.

2.3. Feature extraction

The feature used in this study is mel-frequency cepstral coefficients (MFCC). The use of MFCC is based on research conducted by Sustika *et al*. [34] and Swedia *et al*. [35] which states that MFCC feature input can provide better performance for speech recognition using deep learning methods. MFCC has seven

stages, namely the pre-emphasize process, frame blocking, windowing, fast fourier transform, mel filterbank, discrete cosine transforms, and cepstral liftering [1]. Pre-emphasize is used to amplify energy in the high-frequencies of the input speech signal. Frame blocking is a stage for dividing the speech signal into several frames and there are overlapping parts between frames to avoid missing signal data. In this study, a frame size of 25 milliseconds will be used and an overlap of 10 milliseconds or 40% of the frame length will be used. The windowing process used in this study is the Hamming window function.

Fast Fourier Transform (FFT) will transform the speech signal from the time domain to the frequency domain [3]. The results of the Fast Fourier Transform will be filtered using a mel scale with the help of a triangular filter bank to find out the available energy at each point. In this process 40 filters are used on the mel scale to extract the frequency bands. Discrete cosine transform is the next stage of the MFCC algorithm, which will produce the coefficients used for recognition of a speech signal. The coefficients are obtained by converting the mel-spectrum into the time domain. The number of coefficients produced by this process are 13, 26 and 39. Determination of this value is based on previous research that we have conducted. The value contained in these coefficients is called the acoustic vector which characterizes a sound signal.

2.4. Normalization

The next step after the feature extraction stage is normalization. Feature normalization is performed using the Z-Score Normalization method [36], also known as standardization. This is generated by subtracting the feature coefficient value from the feature mean and then dividing it by the feature standard deviation.

2.5. Model training

The model architecture used in the training process can be seen in Figure 1. In this architecture, the convolution process is done before being processed into Bi-LSTM. There are two blocks in the proposed architecture. The convolution block consists of convolution with ReLU, max pooling, dropout, and batch normalization, while the Bi-LSTM block consists of Bi-LSTM with tanh, dropout, and batch normalization. N indicates the number of layers in Bi-LSTM.

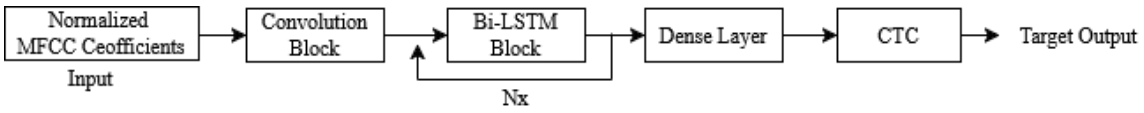


Figure 1. Model architecture

In the convolution block, several kernel and filter sizes are tested so the best model can be selected for the next step of the speech recognition process. Kernels are used for the convolution process and filters are used for the max pooling process. More details on kernel and filter sizes can be seen in the test scenarios subsection. The dropout function is a regularization technique that helps prevent overfitting by randomly setting a fraction of input units to zero during training. Those units (along with their connections) will not contribute to the forward pass or backpropagation during a particular training iteration. Batch normalization is a technique to improve training speed and stability. It involves normalizing the activations of each layer across the mini-batch during training. This normalization helps to mitigate issues like internal covariate shift and ensures that the inputs to each layer remain within a certain range, which can speed up training.

The connectionist temporal classification (CTC) is used to determine the sequence distribution by applying the softmax function to the Dense Layer network result output for each time step [37]. The input is a sequence of observations (such as acoustic features for speech recognition or pen-tip positions for handwriting recognition), and the output is a sequence of symbols (such as phonemes for speech or characters for handwriting). However, the alignment between the input and output sequences may not be one-to-one; there may be many-to-one or many-to-many mappings. CTC addresses this problem by allowing the learning algorithm to learn such mappings directly from the data without requiring explicitly aligned input-output pairs. It works by introducing a "blank" symbol and allowing repeated occurrences of symbols in the output sequence. By allowing the model to learn the alignment between input and output sequences directly from the data, CTC enables end-to-end training of sequence models without the need for handcrafted alignment annotation.

2.6. Testing

The result of the training process is a speech recognition model. Using the model generated during training, the next step is the testing phase. In this phase, the data used consists of test data, which accounts for 20% of the total dataset.

2.7. Evaluation

Evaluation is done by looking at the value of the train loss and validation loss during the training process and calculating the WER during the testing process. Train loss refers to the measure of error between the predicted values of a model during training and the actual values. Validation loss is a metric for evaluating model performance on data that has not been trained. During training, 20% of the dataset is set as validation data. The smaller the loss and WER values, the better the performance. WER calculates the percentage of words that are predicted incorrectly based on three main types of errors, namely, substitution if words are incorrectly transcribed, insertion if the words transcribed by the model are not in the reference text, and deletion if words are deleted during transcription. WER can be calculated using (1) [38].

$$WER = \frac{S+I+D}{N} = \frac{S+I+D}{H+S+D} \quad (1)$$

where S is the total substitution, I is the total insertion, D is the total deletion, N is the total word in reference text, and H is the total word correctly transcripted.

3. RESULTS AND ANALYSIS

3.1. Testing scenarios

There are three scenarios used in this study, as follows:

– Scenario 1. Bi-LSTM without dropout

In this scenario, hyperparameters are as follows:

a. Learning rate: 0.1, 0.01, 0.02, and 0.05
b. Number of units: 32, 64, and 128
c. Number of layers: 1, 2, and 3
d. Number of experiments for each hyperparameter combination: 3 times

The metrics used in this test are the train loss and validation loss during the training process and WER measurements for the best trial test for each hyperparameter combination. The best WER value for the best parameter combination will be extended by adding layers of up to 7 layers.

– Scenario 2. Bi-LSTM with dropout

This scenario was conducted to test the use of dropouts and the effect of Batch Normalization on the model. The experiments conducted in this scenario is performed by using the learning rate, number of hidden units and number of layers resulted from scenario 1. The dropout value used is 0.1; 0.2; 0.3; 0.4; 0.5; 0.6. After getting the best dropout, then this value is used to test several Bi-LSTM layers. The number of layers tested further is 1, 2, 3, 4 and the dropout used is 0.1; 0.2; 0.3; and 0.4.

– Scenario 3. Conv+Bi-LSTM

This scenario is performed to test whether the addition of convolution layer to Bi-LSTM will further increase the accuracy of its recognition rate. One layer convolution is tested to several layers of Bi-LSTM. Some of the hyperparameters involved are as follows:

a. Size of kernels: 3×3, 5×5, 7×7, 9×9, and 11×11
b. Number of filters: 32, 64, 128
c. Number of Bi-LSTM layers: 1, 2, 3, 4, and 5
d. Number of experiments for each hyperparameter combination: 3 times

In this scenario, the best combination is sought by looking at loss and validation loss in the training process and WER in the testing process.

3.2. Experiment result

– Experiment result of scenario 1

Table 1 shows the results of scenario 1. In this table, the training loss, validation loss and WER for each hyperparameter combination that has been previously set are shown. Based on the WER trend of each learning rate and the number of hidden units used, a learning rate of 0.05 was chosen, with a total of 64 hidden units to be used in the next training process. Those values are chosen because the WER decreasing trend is more stable and more promising to produce the lowest WER if additional layers are used. Therefore, further testing was carried out by adding layers at a learning rate of 0.05 and a total of 64 Units. Each parameter combination was carried out 3 times an experiment and the best value was taken. Table 2 shows the test results.

Table 1. Experiment result of scenario 1

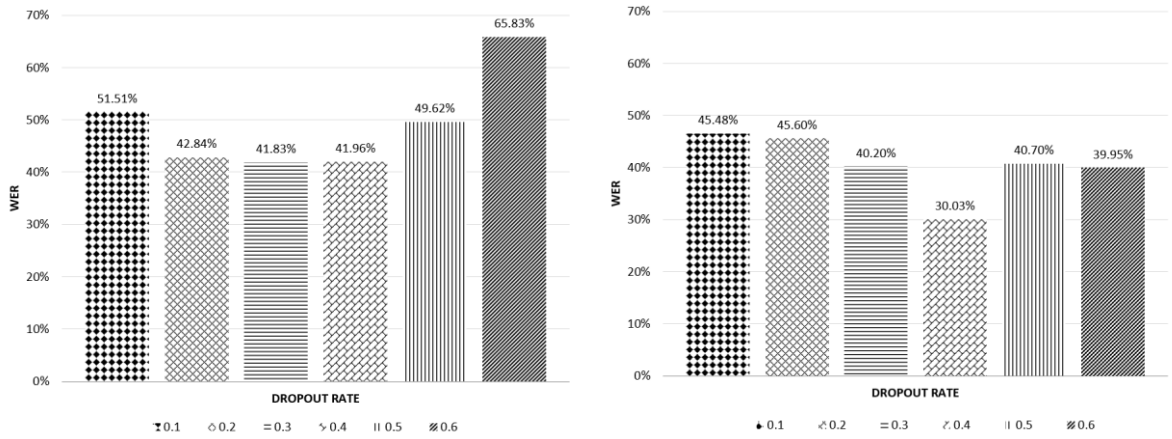
Number of layers	Learning rate	Hidden unit 32 Loss	32 V loss	32 WER (%)	64 Loss	64 V Loss	64 WER (%)	128 Loss	128 V loss	128 WER (%)
1	0.01	35.063	36.912	99.50	36.379	40.363	98.74	19.284	40.245	**95.60**
	0.02	37.110	42.333	98.74	28.938	41.650	97.11	31.705	40.739	97.61
	0.05	45.073	44.990	101.26	30.753	44.997	96.36	26.191	40.523	96.99
	0.1	53.519	50.099	99.87	43.439	49.754	99.50	31.050	37.010	96.99
2	0.01	24.457	40.212	97.99	23.119	42.948	95.98	18.365	45.066	98.74
	0.02	23.983	35.564	94.22	29.767	39.114	98.37	11.729	42.659	95.85
	0.05	26.751	31.606	94.72	15.529	36.261	88.32	17.054	41.062	89.70
	0.1	43.487	42.198	97.11	35.200	32.840	95.23	12.572	29.575	**86.56**
3	0.01	22.864	39.336	93.09	13.873	40.192	94.72	19.372	42.913	95.73
	0.02	18.521	29.921	91.96	30.457	37.236	94.85	21.903	34.388	95.35
	0.05	18.612	28.077	**83.54**	5.057	29.817	**77.14**	4.150	30.351	**78.52**
	0.1	35.537	32.370	96.11	19.913	31.837	86.95	5.583	18.089	**67.59**

Table 2. Effect of addition layer on Bi-LSTM with LR = 0.05 and hidden unit = 64

Number of layers	Loss	V loss	WER (%)
4	5.0189	21.4785	63.07
5	5.7762	15.0152	**54.77**
6	7.3146	15.5222	61.81
7	15.7348	20.6376	72.74

– Experiment result of scenario 2

Based on the results of scenario 1, it can be seen that the smallest WER value is in a model with a learning rate of 0.05, the number of hidden units is 64 and the number of layers = 5. Henceforth these results are used to test the effect of using batch normalization (BN). Figure 2 shows a WER comparison graph to show the effect of BN implemented on the Bi-LSTM architecture. Figure 2(a) shows the WER value when using the Bi-LSTM+dropout model with BN, while Figure 2(b) shows the WER value when using the Bi-LSTM+dropout model without BN. From Figure 2 it can be seen that the smallest WER is obtained in the Bi-LSTM+Dropout model without using BN. With this model, various predefined dropout values are then tested. The results can be seen in Table 3. While the WER value if the Bi-LSTM model does not use a dropout is 54.77%.



(a) (b)

Figure 2. Effect of batch normalization to WER; (a) Bi-LSTM with BN and (b) Bi-LSTM without BN

Table 3. WER value based on the number of layers in the Bi-LSTM+Dropout

Number of Bi-LSTM Layer	WER (%) D=0.1	D=0.2	D=0.3	D=0.4
1	**96.73**	97.61	**96.73**	98.87
2	84.42	83.04	**80.91**	81.03
3	62.06	**54.90**	54.90	55.03
4	46.48	44.60	**44.35**	48.87
5	46.48	45.60	40.20	**30.03**

D: Dropout

– Experiment result of scenario 3

Experiment result of scenario 3 is shown in Table 4. The number of Bi-LSTM layers tested was only up to the fifth layer because based on the results of experiment 1, the addition of a sixth and seventh Bi-LSTM layer produced a higher WER value compared to using five layers.

Table 4. Experiment result of scenario 3

Model	Filter	WER (%)				
		K=11	K=9	K=7	K=5	K=3
CB1	32	97.74	96.23	96.73	96.86	**93.72**
	64	100.13	97.61	96.99	97.74	**96.48**
	128	**96.73**	99.62	97.24	98.24	98.74
CB2	32	82.66	77.39	75.25	74.50	**70.73**
	64	86.68	79.90	77.01	79.71	**69.10**
	128	78.39	78.83	66.06	80.37	**64.30**
CB3	32	**50.63**	54.77	52.51	60.80	61.18
	64	52.51	56.41	**49.25**	54.77	58.42
	128	64.50	59.17	**56.72**	56.86	57.00
CB4	32	35.30	34.17	50.88	**33.29**	49.62
	64	40.45	39.93	34.05	**29.15**	40.20
	128	**35.43**	48.24	36.70	36.81	42.97
CB5	32	35.80	30.40	**28.14**	36.68	45.19
	64	**27.14**	29.15	28.77	35.55	36.94
	128	35.55	**31.03**	36.68	38.57	34.30

K=Kernel
CBx=1 layer Convolution+x layer Bi-LSTM

3.3. Discussion

The results of scenario 1, it is shown that the Bi-LSTM model with 5 layers and 64 units executed using learning 0.05 shows a decrease in WER of 22.37% from Bi-LSTM with 3 layers. Meanwhile, the result of scenario 2 (Table 3) show that the addition of dropouts improves speech recognition performance. It also shows that the Bi-LSTM 2 layer to 4 layers with a dropout rate of 0,3 gives the lowest WER compared to other dropout rates and outperforms the performance of Bi-LSTM without dropouts.

Meanwhile, the Bi-LSTM 1 layer works optimally if a dropout rate of 0,1 is applied, but this result is not better than the Bi-LSTM model without dropouts. So that the Bi-LSTM 1 layer for testing the effect of convolution is not applied dropout. Figure 2 show that the model with the addition of dropouts without applying batch normalization gives better performance than the model with batch normalization.

Based on the results of scenario3, it is proven that convolution in several Bi-LSTM layers is able to improve the performance of language speech recognition Indonesia. The greater the number of layers in Bi-LSTM, the better the WER value. For more details, the decrease in WER which shows the performance of adding convolution to the Bi-LSTM architecture can be seen in Table 5.

Table 5. Performance of adding convolution to Bi-LSTM architecture

Number of Bi-LSTM layer	WER (%)				Improvement (%)
	Bi-LSTM	Bi-LSTM+Dropout	Conv+Bi-LSTM+Dropout	Decrease of WER	
1	95.60	96.73	93.72	3.01	3.11
2	86.56	80.91	64.30	16.61	20.53
3	67.59	54.90	49.25	5.65	10.29
4	63.07	44.35	29.15	15.2	34.27
5	54.77	30.03	27.14	2.89	9.63
Average					**15.56**

Compared with other studies, the results of this study can compete with other methods for Indonesian speech recognition. Figure 3 shows the WER values for the LSTM [7], Bi-LSTM, and Conv+Bi-LSTM methods. From Figure 3, it can be observed that Conv+Bi-LSTM yields a lower WER compared to LSTM and Bi-LSTM. It indicates that speech recognition using Conv+Bi-LSTM performs better than Bi-LSTM or LSTM alone.

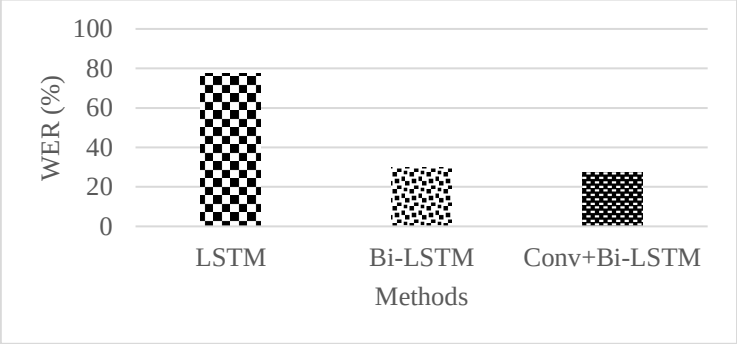


Figure 3. Comparison with other methods

4. CONCLUSION

Optimization of Indonesian continuous speech recognition is performed by adding convolutions to the Bi-LSTM architecture. The results showed that by adding a convolution layer to Bi-LSTM architectures, speech recognition performance is significantly improved with an average WER reduction of 15.56%. In future work, these results can still be improved by adding input features with delta coefficients (first-order derivatives (or differences) of MFCC coefficients) and delta-delta coefficients (second-order derivatives of MFCC coefficients) or by increasing the number of convolution layers.

FUNDING INFORMATION

The authors would like to acknowledge the research funding supported by Faculty of Science and Mathematics, Universitas Diponegoro under the Grant of Intermediate Research – Contract Number 1264F/UN7.5.8/PP/2022.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Sukmawati Nur Endah	✓	✓		✓	✓	✓		✓	✓	✓		✓	✓	✓
Rismiyati		✓	✓			✓			✓	✓			✓	✓
Priyo Sidik Sasongko	✓				✓		✓			✓	✓	✓		✓
Anwar Petrus F. Naiborhu			✓	✓		✓	✓	✓	✓		✓			

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors declare that we have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper. Therefore, authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author, [SNE], upon reasonable request.

REFERENCES

[1] A. Mehrish, N. Majumder, R. Bhardwaj, R. Mihalcea, and S. Poria, "A Review of Deep Learning Techniques for Speech Processing," *Information Fusion*, vol. 99, no. 101869, 2023, doi: 10.1016/j.inffus.2023.101869.

[2] A. S. Ganakwar, S. K. Maher, and R. R. Deshmukh, "Enhancing Sanskrit Isolated Word Recognition: A Comparative Analysis of MFCC and SVM Feature Integration," in *2023 26th Conference of the Oriental COCOSDA International Committee for the Co-ordination and Standardisation of Speech Databases and Assessment Techniques (O-COCOSDA)*, India, 2023, doi: 10.1109/O-COCOSDA60357.2023.10482969.

[3] S. N. Endah, N. Fadlilah, R. Kusumaningrum, and S. Adhy, "Continuous Speech Segmentation Using Dynamic Thresholding of Short-term Features," *Journal of Engineering Science and Technology*, vol. 17, no. 4, pp. 2919 - 2935, 2022.

[4] J.-U. Bang *et al*., "KsponSpeech: Korean Spontaneous Speech Corpus for Automatic Speech Recognition," *Applied Sciences*, vol. 10, no. 19 : 6939, 2020, doi: 10.3390/app10196936.

[5] Z. Hatala, "Speech recognition for Indonesian language and its application to home automation," in *2019 4th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, Yogyakarta, Indonesia, 2019, pp. 323-328, 2019, doi: 10.1109/ICITISEE48480.2019.9003806.

[6] H. Z. Muhammad, M. Nasrun, C. Setianingsih, and M. A. Murti, "Speech recognition for English to Indonesian translator using Hidden Markov Model," in *2018 International Conference on Signals and Systems (ICSigSys)*, Bali, Indonesia, pp. 255-260, 2018, doi: 10.1109/ICSIGSYS.2018.8372768.

[7] A. P. F. Naiborhu and S. N. Endah, "Indonesian Continuous Speech Recognition Using CNN and Bidirectional LSTM," in *2021 5th International Conference on Informatics and Computational Sciences (ICICoS)*, Semarang, 2021, doi: 10.1109/ICICoS53627.2021.9651902.

[8] S. Suyanto, A. Arifianto, A. Sirwan, and A. P. Rizaenra, "End-to-End Speech Recognition Models for a Low-Resourced Indonesian Language," in *2020 8th International Conference on Information and Communication Technology (ICoICT)*, Yogyakarta, Indonesia, 2020, doi: 10.1109/ICoICT49345.2020.9166346.

[9] T. F. Abidin, A. Misbullah, R. Ferdhiana, M. Z. Aksana, and L. Farsiah, "Deep Neural Network for Automatic Speech Recognition from Indonesian Audio using Several Lexicon Types," in *2020 International Conference on Electrical Engineering and Informatics (ICELTICs)*, Aceh, Indonesia, pp. 1-5, 2020, doi: 10.1109/ICELTICs50595.2020.9315538.

[10] A. Sirwan, K. A. Thama, and S. Suyanto, "Indonesian Automatic Speech Recognition Based on End-to-end Deep Learning Model," in *2022 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom)*, Malang, Indonesia, pp. 410-415, 2022, doi: 10.1109/CyberneticsCom55287.2022.9865253.

[11] R. Yang, J. Yang, and Y. Lu, "Indonesian speech recognition based on Deep Neural Network," in *2021 International Conference on Asian Language Processing (IALP)*, Singapore, 2021, doi: 10.1109/IALP54817.2021.9675280.

[12] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations," in *34th Conference on Neural Information Processing Systems* , Vancouver, Canada, 2020.

[13] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. Mcleavey, and I. Sutskever, "Robust Speech Recognition via Large-Scale Weak Supervision," in *Proceedings of the 40th International Conference on Machine Learning*, Honolulu, Hawaii, 2023.

[14] J. Li, "Recent Advances in End-to-End Automatic Speech Recognition," *APSIPA Transactions on Signal and Information Processing*, vol. 11, no. e 8, 2022, doi: 10.1561/116.00000050.

[15] T. Hori, J. Cho and S. Watanabe, "End-to-end Speech Recognition With Word-Based RNN Language Models," in *IEEE Spoken Language Technology Workshop (SLT)*, Athens, Greece, 2018, doi: 10.1109/SLT.2018.8639693.

[16] J. Li, R. Zhao, H. Hu and Y. Gong, "Improving RNN Transducer Modeling for End-to-End Speech Recognition," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Singapore, 2019, doi: 10.1109/ASRU46091.2019.9003906.

[17] J. Li, A. Mohamed, G. Zweig and Y. Gong, "Exploring multidimensional lstms for large vocabulary ASR," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, China, 2016, doi: 10.1109/ICASSP.2016.7472617.

[18] F. Weninger *et al*., "Speech Enhancement with LSTM Recurrent Neural Networks and its Application to Noise-Robust ASR," in *Latent Variable Analysis and Signal Separation. LVA/ICA 2015. Lecture Notes in Computer Science, vol 9237*, Switzerland, Springer, 2015, pp. 91-99, doi: 10.1007/978-3-319-22482-4_11.

[19] A. B. Gumelar, E. M. Yuniarno, D. P. Adi, A. G. Sooai, I. Sugiarto, and M. H. Purnomo, "BiLSTM-CNN Hyperparameter Optimization for Speech Emotion and Stress Recognition," in *International Electronics Symposium (IES)*, Surabaya, Indonesia, 2021, doi: 10.1109/IES53407.2021.9594024.

[20] L. Dong, S. Xu, and B. Xu, "Speech-transformer: A No-recurrence Sequence-to-sequence Model For Speech Recognition," in *International Conference on Acoustics, Speech, and Signal Processing*, Calgary, Alberta, Canada, 2018, doi: 10.1109/ICASSP.2018.8462506.

[21] S. Kim, *et al.*, "Squeezeformer: An Efficient Transformer for Automatic Speech Recognition," in *Proceedings of Advances in Neural Information Processing Systems 35*, New Orleans, 2022, doi: 10.48550/arXiv.2206.00888

[22] A. Gulati, *et al.*, "Conformer: Convolution-augmented Transformer for Speech Recognition," in *Interspeech*, Shanghai, Tiongkok, 2020, doi: 10.48550/arXiv.2005.08100.

[23] S. Li, M. Xu, and X.-L. Zhang, "Efficient conformer-based speech recognition with linear attention," 2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Tokyo, Japan, 2021, doi: 10.48550/arXiv.2104.06865.

[24] S. Karita, *et al*., "A Comparative Study on Transformer vs RNN in Speech Applications," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Singapore, 2019, doi: 10.1109/ASRU46091.2019.9003750.

[25] A. Sherstinsky, "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network," *Physica D: Nonlinear Phenomena*, vol. 404, no. 132306, 2020, doi: 10.1016/j.physd.2019.132306

[26] A. Shewalkar, D. Nyavanandi, and S. A. Ludwig, "Performance Evaluation of Deep neural networks Applied to Speech Recognition: RNN, LSTM and GRU," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 9, no. 4, p. 235–245, 2019, doi: 10.2478/jaiscr-2019-0006.

[27] D. Wang, S. Lv, X. Wang, and X. Lin, "Gated Convolutional LSTM for Speech Commands Recognition," in *Computational Science – International Conference on Computational Science 2018. Lecture Notes in Computer Science*, 2018, doi: 10.1007/978-3-319-93701-4_53.

[28] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech Recognition with Deep Recurrent Neural Networks," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Canada, 2013, doi: 10.1109/ICASSP.2013.6638947.

[29] A. Graves, N. Jaitly, and A.-r. Mohamed, "Hybrid Speech Recognition With Deep Bidirectional LSTM," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, Olomouc, Czech Republic, 2013, doi: 10.1109/ASRU.2013.6707742.

[30] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks*, vol. 18, no. 5, pp. 602-610, 2005, doi: 10.1016/j.neunet.2005.06.042.
[31] A. Graves, S. Fernandez, and J. Schmidhuber, "Bidirectional LSTM Networks for Improved Phoneme Classification and Recognition. In: Duch, Artificial Neural Networks: Formal Models and Their Applications – ICANN 2005," in *Lecture Notes in Computer Science volume 3697*, Berlin, Springer-Verlag Berlin Heidelberg, 2005, p. 799–804, doi: 10.1007/11550907_126.
[32] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language Modeling with Gated Convolutional Networks," in *Proceedings of the 34th International Conference on Machine Learning*, Sydney, Australia, 2017.
[33] N. Bensalah, H. Ayad, A. Adib, and A. I. E. Farouk, "CRAN: An Hybrid CNN-RNN Attention-Based Model for Arabic Machine Translation," in *Networking, Intelligent Systems and Security. Smart Innovation, Systems and Technologies*, vol 237, pp 87-102, 2022, doi : 10.1007/978-981-16-3637-0_7.
[34] R. Sustika, A. R. Yuliani, E. Zaenudin, and H. F. Pardede, "On comparison of deep learning architectures for distant speech recognition," Yogyakarta, Indonesia, 2017, doi: 10.1109/ICITISEE.2017.8285488.
[35] E. R. Swedia, A. B. Mutiara, M. Subali and Ernastuti, "Deep learning long-short term memory (LSTM) for Indonesian speech digit recognition using LPC and MFCC Feature," Palembang, Indonesia, 2018, pp. 1-5, doi: 10.1109/IAC.2018.8780566.
[36] D. Singh and B. Singh , "Investigating the impact of data normalization on classification performance," *Applied Soft Computing Journal*, vol. 97, no. 105524, 2020, doi: 10.1016/j.asoc.2019.105524.
[37] S. S. Alrumiah and A. A. Al-Sharqabi, "Intelligent Quran Recitation Recognition and Verification: Research Trends and Open Issues," *Arabian Journal for Science and Engineering* , Springer, 2022, doi: 10.1007/s13369-022-07273-8.
[38] S. K. Gaikwad, B. W. Gawali, and P. Yannawar, "A Review on Speech Recognition Technique," *International Journal of Computer Applications*, vol. 10, no. 3, pp. 16-24, 2010.

BIOGRAPHIES OF AUTHORS

Sukmawati Nur Endah received the B.Sc. degree in Mathematical from the Universitas Diponegoro, Semarang, Indonesia, in 2001, and the Master's degree in Computer Science from the Universitas Indonesia, Depok, Indonesia in 2009, respectively. She is currently a lecturer in Department of Informatics, Universitas Diponegoro. In 2019-2022, she has been the Head of the Laboratorium of Intelligent Systems in the Department of Informatics, Universitas Diponegoro. Her current research interests include Intelligent system, natural language processing, speech recognition, and machine learning. She can be contacted at email: sukmane@lecturer.undip.ac.id.

Rismiyati received the B.Eng. degree from the Nanyang Technological University, Singapore in 2007 and the the Master's degree in Computer Science from the Universitas Gadjah Mada, Indonesia in 2016. She is currently working as a lecturer with the Department of Informatics, Universitas Diponegoro, Indonesia. Her main research interests are machine learning and computer vision. She can be contacted at email: rismiyati@live.undip.ac.id.

Priyo Sidik Sasongko received the B.Sc. degree in Mathematical from the Universitas Diponegoro, Semarang, Indonesia, in 1996, and the Master's degree in Computer Science from the Universitas Gadjah Mada, Yogyakarta, Indonesia in 2006, respectively. He is currently working as a lecturer with the Department of Informatics, Universitas Diponegoro, Indonesia. His main research interests are machine learning and artificial intelligence. He can be contacted at email: priyosidiksasongko@lecturer.undip.ac.id.

Anwar Petrus F. Naiborhu received the B.Sc. degree in Computer Science from the Universitas Diponegoro, Semarang, Indonesia, in 2021. He has been a research assistant in the Laboratory of Intelligent System in Department of Informatics, Universitas Diponegoro since 2020. His main research interests are machine learning and speech recognition. He can be contacted at email: naiborhupetrus@alumni.undip.ac.id.